Automatically Creating Bilingual Lexicons for Machine Translation from Bilingual Text

Davide Turcato

Natural Language Lab
School of Computing Science
Simon Fraser University
Burnaby, BC, V5A 1S6
Canada
turk@cs.sfu.ca

TCC Communications
100-6722 Oldfield Road
Victoria, BC
V8M 2A3
Canada
turk@tcc.bc.ca

Abstract

A method is presented for automatically augmenting the bilingual lexicon of an existing Machine Translation system, by extracting bilingual entries from aligned bilingual text. The proposed method only relies on the resources already available in the MT system itself. It is based on the use of bilingual lexical templates to match the terminal symbols in the parses of the aligned sentences.

1 Introduction

A novel approach to automatically building bilingual lexicons is presented here. The term *bilingual lexicon* denotes a collection of complex equivalences as used in Machine Translation (MT) transfer lexicons, not just word equivalences. In addition to words, such lexicons involve syntactic and semantic descriptions and means to perform a correct transfer between the two sides of a bilingual lexical entry.

A symbolic, rule-based approach of the *parse-parse-match* kind is proposed. The core idea is to use the resources of bidirectional transfer MT systems for this purpose, taking advantage of their features to convert them to a novel use. In addition to having them use their bilingual lexicons to produce translations, it is proposed to have them use translations to produce bilingual lexicons. Although other uses might be conceived, the most appropriate use is to have an MT system automatically augment its own bilingual lexicon from a small initial sample.

The core of the described approach consists of using a set of bilingual lexical templates in matching the parses of two aligned sentences and in turning the lexical equivalences thus established into new bilingual lexical entries.

2 Theoretical framework

The basic requirement that an MT system should meet for the present purpose is to be *bidirectional*. Bidirectionality is required in order to ensure that both source and target grammars can be used for parsing and that transfer can be done in both directions. More precisely, what is relevant is that the input and output to transfer be the same kind of structure.

Moreover, the proposed method is most productive with a lexicalist MT system (Whitelock, 1994). The proposed application is concerned with producing bilingual lexical knowledge and this sort of knowledge is the only type of bilingual knowledge required by lexicalist systems. Nevertheless, it is also conceivable that the present approach can be used with a non-lexicalist transfer system, as long as the system is bidirectional. In this case, only the lexical portion of the bilingual knowledge can be automatically produced, assuming that the structural transfer portion is already in place. In the rest of this paper, a lexicalist MT system will be assumed and referred to. For the specific implementation described here and all the examples, we will refer to an existing lexicalist English-Spanish MT system (Popowich et al., 1997).

The main feature of a lexicalist MT system is that it performs no structural transfer. Transfer is a mapping between a bag of lexical items used in parsing (the *source bag*) and a corresponding bag of target lexical items (the *target bag*), to be used in generation. The source bag actually contains more information than the corresponding bag of lexical items before parsing. Its elements get enriched with additional information instantiated during the parsing process. Information of fundamental importance included therein is a system of indices that express de-

pendencies among lexical items. Such dependencies are transferred to the target bag and used to constrain generation. The task of generation is to find an order in which the lexical items can be successfully parsed.

3 Bilingual templates

A *bilingual template* is a bilingual entry in which words are left unspecified. E.g.:

(1)
```
_ ::  (L,@count_noun(A)) ↔
_ ::  (R,@noun(A))
\\trans_noun(L,R).
```

Here, a '`::`' operator connects a word (a variable, in a template) to a description, '↔' connects the left and right sides of the entry, '\\' introduces a *transfer macro*, which takes two descriptions as arguments and performs some additional transfer (Turcato et al., 1997). Descriptions are mainly expressed by macros, introduced by a '`@`' operator. The macro arguments are indices, as used in lexicalist transfer.

Templates have been widely used in MT (Buschbeck-Wolf and Dorna, 1997), particularly in the Example-Based Machine Translation (EBMT) framework (Kaji et al. (1992), Güvenir and Tunç (1996)). However, in EBMT, templates are most often used to model sentence-level correspondences, rather then lexical equivalences. Consequently, in EBMT the relation between lexical equivalences and templates is the reverse of what is being proposed here. In EBMT, lexical equivalences are assumed and (sentential) templates are inferred from them. In the present framework, sentential correspondences (in the form of possible combinations of lexical templates) are assumed and lexical equivalences are inferred from them.

In a lexicalist approach, the notion of bilingual lexical entry, and thus that of bilingual template, must be intended broadly. Multiword entries can exist. They can express dependencies among lexical items, thus being suitable for expressing phrasal equivalences. In brief, bilingual lexical entries can exhaustively cover all the bilingual information needed in transfer.

In a lexicalist MT system, transfer is accomplished by finding a bag of bilingual entries partitioning the source bag. The source side of each entry (in the rest of this paper: the left hand side) corresponds to a cell of the partition. The union of the target sides of the entries constitutes the target bag. E.g.:

(2) a. Source bag:
$\{Sw_1{::}Sd_1,\ Sw_2{::}Sd_2,\ Sw_3{::}Sd_3\}$

b. Bilingual entries:
$\{Sw_1{::}Sd_1\ \&\ Sw_3{::}Sd_3 \leftrightarrow$
$Tw_1{::}Td_1\ \&\ Tw_2{::}Td_2,$

$Sw_2{::}Sd_2 \leftrightarrow$
$Tw_3{::}Td_3\ \&\ Tw_4{::}Td_4\}$

c. Target bag:
$\{Tw_1{::}Td_1,\ Tw_2{::}Td_2,\ Tw_3{::}Td_3,$
$Tw_4{::}Td_4\}$

where each $Sw_i{::}Sd_i$ and $Tw_i{::}Td_i$ are, respectively, a source and target $<Word,Description>$ pair. In addition, the bilingual entries must satisfy the constraints expressed by indices in the source and target bags. The same information can be used to find (2b), given (2a) and (2c).

Any bilingual lexicon is partitioned by a set of templates. The entries in each equivalence class only differ by their words. A bilingual lexical entry can thus be viewed as a triple $<Sw,Tw,T>$, where Sw is a list of source words, Tw a list of target words, and T a template. A set of such bilingual templates can be intuitively regarded as a 'transfer grammar'. A grammar defines all the possible sequences of pre-terminal symbols, i.e. all the possible types of sentences. Analogously, a set of bilingual templates defines all the possible translational equivalences between bags of pre-terminal symbols, i.e. all the possible equivalences between types of sentences.

Using this intuition, the possibility is explored of analyzing a pair of such bags by means of a database of bilingual templates, to find a bag of templates that correctly accounts for the translational equivalence of the two bags, without resorting to any information about words. In the example (2), the following bag of templates would be the requested solution:

(3) $\{\_{::}Sd_1\ \&\ \_{::}Sd_3 \leftrightarrow \_{::}Td_1\ \&\ \_{::}Td_2,$

$\_{::}Sd_2 \leftrightarrow \_{::}Td_3\ \&\ \_{::}Td_4\}$

Equivalences between (bags of) words are automatically obtained as a result of the process, whereas in translating they are assumed and used to select the appropriate bilingual entries.

Templates	Entries	Coverage
1	5683	33.9 %
2	8726	52.1 %
3	10710	63.9 %
4	12336	73.6 %
5	13609	81.2 %
50	15473	92.3 %
500	16338	97.5 %
922	16760	100.0 %

Table 1: Incremental template coverage

The whole idea is based on the assumption that a lexical item's description and the constraints on its indices are sufficient in most cases to uniquely identify a lexical item in a parse output bag. Although exceptions could be found (most notably, two modifiers of the same category modifying the same head), the idea is viable enough to be worth exploring.

The impression might arise that it is difficult and impractical to have a set of templates available in advance. However, there is empirical evidence to the contrary. A count on the MT system used here showed that a restricted number of templates covers a large portion of a bilingual lexicon. Table 1 shows the incremental coverage. Although completeness is hard to obtain, a satisfactory coverage can be achieved with a relatively small number of templates.

In the implementation described here, a set of templates was extracted from the MT bilingual lexicon and used to bootstrap further lexical development. The whole lexical development can be seen as an interactive process involving a bilingual lexicon and a template database. Templates are initially derived from the lexicon, new entries are successively created using the templates. Iteratively, new entries can be manually coded when the automatic procedure is lacking appropriate templates and new templates extracted from the manually coded entries can be added to the template database.

4 The algorithm

In this section the algorithm for creating bilingual lexical entries is described, along with a sample run. The procedure was implemented in Prolog, as was the MT system at hand. Basically, a set of lexical entries is obtained from a pair of sentences by first parsing the source and target sentences. The source bag is then transferred using templates as transfer rules (plus entries for closed-class words and possibly a pre-existing bilingual lexicon). The transfer output bag is then unified with the target sentence parse output bag. If the unification succeeds, the relevant information (bilingual templates and associated words) is retrieved to build up the new bilingual entries. Otherwise, the system backtracks into new parses and transfers.

The main predicate `make_entries/3` matches a source and a target sentence to produce a set of bilingual entries:

```
make_entries(Source,Target,Entries):-
   parse_source(Source,Deriv1),
   parse_target(Target,Deriv2),
   transfer(Deriv1,Deriv3),
   get_bag(Deriv2,Bag2),
   get_bag(Deriv3,Bag3),
   match_bags(Bag2,Bag3,Bag4),
   get_bag(Deriv1,Bag1),
   make_be_info(Bag1,Bag4,Deriv3,Be),
   be_info_to_entries(Be,Entries).
```

Each `Deriv`n variable points to a buffer where all the information about a specific derivation (parse or transfer) is stored and each `Bag`n variable refers to a bag of lexical items. Each step will be discussed in detail in the rest of the section. A sample run will be shown for the following English-Spanish pair of sentences:

(4) a. `the fat man kicked out the black dog.`
    b. `el hombre gordo echó el perro negro.`

In the sample session no bilingual lexicon was used for content words. Only a bilingual lexicon for closed class words and a set of bilingual templates were used. Therefore, new bilingual entries were obtained for all the content words (or phrases) in the sentences.

4.1 Source sentence parse

The parse of the source sentence is performed by `parse_source/2`. The parse tree is shown in Fig. 1. Since only lexical items are relevant for the present purposes, only pre-terminal nodes in the tree are labeled.

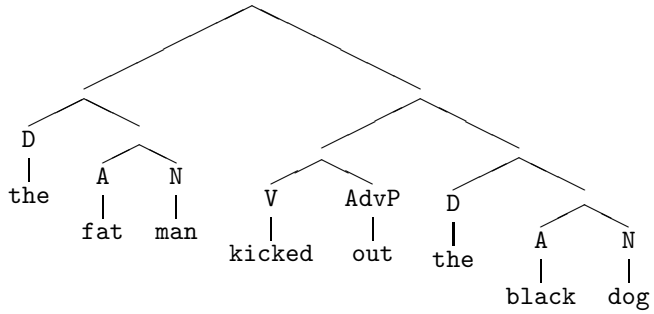

Figure 1: Source sentence parse tree.

Id	Word	Cat	Indices
1	the	determiner	[0]
2	fat	adjective	[0]
3	man	noun	[0]
4	kick	trans_verb	[10,0,9]
5	out	advparticle	[10]
6	the	determiner	[9]
7	black	adjective	[9]
8	dog	noun	[9]

Figure 2: Source sentence parse output bag.

Fig. 2 shows, in succint form, the relevant information from the source bag, i.e. the bag resulting from parsing the source sentence. All the syntactic and semantic information has been omitted and replaced by a category label. What is relevant here is the way the indices are set, as a result of parsing. The words {the,fat,man} are tied together and so are {kick,out} and {the,black,dog}. Moreover, the indices of 'kick' show that its second index is tied to its subject, {the,fat,man}, and its third index is tied to its object, {the,black,dog}.

4.2 Target sentence parse

The parse of the target sentence is performed by parse_target/2. Fig. 3 and 4 show, respectively, the resulting tree and bag. In an analogous manner to what is seen in the source sentence, {el,hombre,gordo} and {el,perro,negro} are, respectively, the subject and the object of 'echó'.

4.3 Transfer

The result of parsing the source sentence is used by transfer/2 to create a translationally equivalent target bag. Fig. 5 shows the result. Transfer is performed by consulting a bilingual lexicon, which, in the present case, contained entries for closed class words (e.g. an entry mapping 'the' to 'el') and templates for content words. The templates relevant to our example are the following:

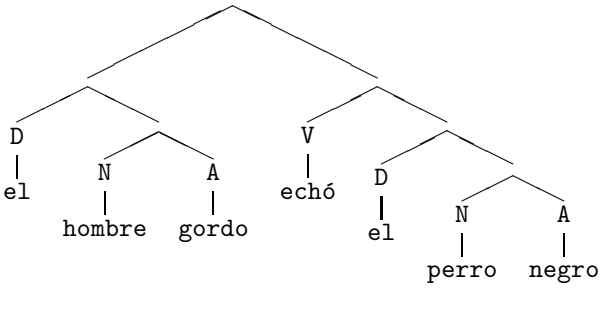

Figure 3: Target sentence parse tree.

Id	Word	Cat	Indices
1	el	d	[0]
2	hombre	n	[0]
3	gordo	adj	[0]
4	echar	v	[1,0,13]
5	el	d	[13]
6	perro	n	[13]
7	negro	adj	[13]

Figure 4: Target sentence parse output bag.

(5) a. _ ::@adj(A)
↔ 'word(adj/adj,1)' ::@adj(A).

b. _ ::(L,@count_noun(A))
↔ 'word(cn/n,1)' ::(R,@noun(A))
\\trans_noun(L,R).

c. _ ::(L,@trans_verb(A,B,C))
& _ ::@advparticle(A)
↔
'word(tv+adv/tv,1)' ::
(R,@verb_acc(A,B,C))
\\trans_verb(L,R).

Id	Word	Cat	Indices
2-1	el	d	[A]
3-2	word(adj/adj,1)	adj	[A]
4-3	word(cn/n,1)	n	[A]
1-4	word(tv+adv/tv,1)	v	[B,A,I]
5-6	el	d	[I]
6-7	word(adj/adj,1)	adj	[I]
7-8	word(cn/n,1)	n	[I]

Figure 5: Transfer output bag.

Bilingual templates are simply bilingual entries with words replaced by variables. Actually, on the target side, words are replaced by labels of the form `word(Ti,Position)`, where `Ti` is a template identifier and `Position` identifies the position of the item in the right hand side of the template. Thus, a label `word(adj/adj,1)` identifies the first word on the right hand side of the template that maps an adjective to an adjective. Such labels are just implementational technicalities that facilitate the retrieval of the relevant information when a lexical entry is built up from a template, but they have no role in the matching procedure. For the present purposes they can entirely be regarded as anonymous variables that can unify with anything, exactly like their source counterparts.

After transfer, the instances of the templates used in the process are coindexed in some way, by virtue of their unification with the source bag items. This is analogous to what happens with bilingual entries in the translation process.

4.4 Target bag matching

The predicate `get_bag/2` retrieves a bag of lexical items associated with a derivation. Therefore, `Bag2` and `Bag3` will contain the bags of lexical items resulting, respectively, from parsing the target sentence and from transfer.

The crucial step is the matching between the transfer output bag and the target sentence parse output bag. The predicate `match_bags/3` tries to unify the two bags (returning the result in `Bag4`). A successful unification entails that the parse and transfer of the source sentence are consistent with the parse of the target sentence. In other words, the bilingual rules used in transfer correctly map source lexical items into target lexical items. Therefore, the lexical equivalences newly established through this process can be asserted as new bilingual entries.

In the matching process, the order in which the elements are listed in the figures is irrelevant, since the objects at hand are bags, i.e. unordered collections. A successful match only requires the existence of a one-to-one mapping between the two bags, such that:

(i) the respective descriptions, here represented by category labels, are unifiable;

(ii) a further one-to-one mapping between the indices in the two bags is induced.

The following mapping between the transfer output bag (Fig. 5) and the target sentence parse output bag (Fig. 4) will therefore succeed:

{<2-1,1>,<3-2,3>,<4-3,2>,<1-4,4>,
<5-6,5>,<6-7,7>,<7-8,6>}

In fact, in addition to correctly unifying the descriptions, it induces the following one-to-one mapping between the two sets of indices:

{<A,0>,<B,1>,<I,13>}

4.5 Bilingual entries creation

The rest of the procedure builds up lexical entries for the newly discovered equivalences and is implementation dependent. First, the source bag is retrieved in `Bag1`. Then, `make_be_info/4` links together information from the source bag, the target bag (actually, its unification with the target sentence parse bag) and the transfer derivation, to construct a list of terms (the variable `Be`) containing the information to create an entry. Each such term has the form `be(Sw,Tw,Ti)`, where `Sw` is a list of source words, `Tw` is a list of target words and `Ti` is a template identifier. In our example, the following `be/3` terms are created:

(6) a. `be([fat],[gordo],adj/adj)`
b. `be([man],[hombre],cn/n)`
c. `be([kick,out],[echar],tv+adv/tv)`
d. `be([black],[negro],adj/adj)`
e. `be([dog],[perro],cn/n)`

Each `be/3` term is finally turned into a bilingual entry by the predicate `be_info_to_entries/2`. The following bilingual entries are created:

(7) a. `fat ::@adj(A)`
↔ `gordo ::@adj(A).`

b. `man ::(D,@count_noun(C))`
↔ `hombre ::(B,@noun(C))`
`\\trans_noun(D,B).`

c. `kick ::(I,@trans_verb(F,G,H))`
`& out ::@advparticle(F)`
↔
`echar ::(E,@verb_acc(F,G,H))`
`\\trans_verb(I,E).`

d. `black ::@adj(J)`
   ↔ `negro ::@adj(J).`

e. `dog ::(M,@count_noun(L))`
   ↔ `hombre ::(K,@noun(L))`
   `\\trans_noun(M,K).`

If a pre-existing bilingual lexicon is in use, bilingual entries are prioritized over bilingual templates. Consequently, only new entries are created, the others being retrieved from the existing bilingual lexicon. Incidentally, it should be noted that a new entry is an entry which differs from any existing entry on either side. Therefore, different entries are created for different senses of the same word, as long as the different senses have different translations.

5 Shortcomings and future work

In matching a pair of bags, two kinds of ambiguity could lead to multiple results, some of which are incorrect. Firstly, as already mentioned, a bag could contain two lexical items with unifiable descriptions (e.g. two adjectives modifying the same noun), possibly causing an incorrect match. Secondly, as the bilingual template database grows, the chance of overlaps between templates also grows. Two different templates or combinations of templates might cover the same input and output. A case in point is that of a phrasal verb or an idiom covered by both a single multi-word template and a compositional combination of simpler templates.

As both potential sources of error can be automatically detected, a first step in tackling the problem would be to block the automatic generation of the entries involved when a problematic case occurs, or to have a user select the correct candidate. In this way the correctness of the output is guaranteed. The possible cost is a lack of completeness, when no user intervention is foreseen.

Furthermore, techniques for the automatic resolution of template overlaps are under investigation. Such techniques assume the presence of a bilingual lexicon. The information contained therein is used to assign preferences to competing candidate entries, in two ways.

Firstly, templates are probabilistically ranked, using the existing bilingual lexicon to estimate probabilities. When the choice is between single entries, the ranking can be performed by counting the frequency of each competing template in the lexicon. The entry with the most frequent template is chosen.

Secondly, heuristics are used to assign preferences, based on the presence of pre-existing entries related in some way to the candidate entries. This technique is suited for resolving ambiguities where multiple entries are involved. For instance, given the equivalence between '`kick the bucket`' and '`estirar la pata`', and the competing candidates

(8) a. {`kick & bucket ↔ estirar & pata`}
    b. {`kick ↔ estirar, bucket ↔ pata`}

the presence of an entry '`bucket ↔ balde`' in the bilingual lexicon might be a clue for preferring the idiomatic interpretation. Conversely, if the hypothetical entry '`bucket ↔ pata`' were already in the lexicon, the compositional interpretation might be preferred.

Finally, efficiency is also dependant on the restrictiveness of grammars. The more grammars overgenerate, the more the combinatoric indeterminacy in the matching process increases. However, overgeneration is as much a problem for translation as for bilingual generation. In other words, no additional requirement is placed on the MT system which is not independently motivated by translation alone.

6 Conclusion

The *parse-parse-match* approach to automatically building bilingual lexicons in not novel. Proposals have been put forward, e.g., by Sadler and Vendelmans (1990) and Kaji et al. (1992).

Wu (1995) points out some possible difficulties of the parse-parse-match approach. Among them, the facts that "appropriate, robust, monolingual grammars may not be available" and "the grammars may be incompatible across languages" (Wu, 1995, 355). More generally, in bilingual lexicon development there is a tendency to minimize the need for linguistic resources specifically developed for the purpose. In this view, several proposals tend to use statistical, knowledge-free methods, possibly in combination with the use of existing Machine Readable Dictionaries (see, e.g., Klavans and Tzoukermann (1995), which also contains a survey of related proposals, pages 195–196).

The present proposal tackles the problem from a different and novel perspective. The acknowledgment that MT is the main application domain to which bilingual resources are relevant is taken as a starting point. The existence of an MT system, for which the bilingual lexicon is intended, is explicitly assumed. The potential problems due to the need for linguistic resources are by-passed by having the necessary resources available in the MT system. Rather than doing away with linguistic knowledge, the pre-existing resources of the pursued application are utilized.

An approach like the present can be most effectively adopted to develop tools allowing MT systems to automatically build their own bilingual lexicons. A tool of this sort would use no extra resources in addition to those already available in the MT system itself. Such a tool would take a small sample of a bilingual lexicon and use it to bootstrap the automatic development of a large lexicon. It is worth noting that the bilingual pairs thus produced would be complete bilingual entries that could be directly incorporated in the MT system, with no post-editing or addition of information.

The only requirement placed by the present approach on MT systems is that they be bi-directional. Therefore, although aimed at the development of specific applications for specific MT systems, the approach is general enough to apply to a wide range of MT systems.

Acknowledgements

This research was supported by TCC Communications, by a Collaborative Research and Development Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC), and by the Institute for Robotics and Intelligent Systems. The author would like to thank Fred Popowich and John Grayson for their comments on earlier versions of this paper.

References

B. Buschbeck-Wolf and M. Dorna. 1997. Using hybrid methods and resources in semantic-based transfer. In *Proceedings of the International Conference 'Recent Advances in Natural Language Processing'*, pages 104–111, Tzigov Chark, Bulgaria.

H. A. Güvenir and A. Tunç. 1996. Corpus-based learning of generalized parse tree rules for translation. In G. McCalla, editor, *Advances in Artificial Intelligence — 11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 121–132. Springer, Berlin.

H. Kaji, Y. Kida, and Y. Morimoto. 1992. Learning translation templates from bilingual text. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 672–678, Nantes, France.

J. Klavans and E. Tzoukermann. 1995. Combining corpus and machine-readable dictionary data for building bilingual lexicons. *Machine Translation*, 10:185–218.

F. Popowich, D. Turcato, O. Laurens, P. McFetridge, J. D. Nicholson, P. McGivern, M. Corzo-Pena, L. Pidruchney, and S. MacDonald. 1997. A lexicalist approach to the translation of colloquial text. In *Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 76–86, Santa Fe, New Mexico, USA.

V. Sadler and R. Vendelmans. 1990. Pilot implementation of a bilingual knowledge bank. In *Proceedings of the 13th International Conference on Computational Linguistics*, pages 449–451, Helsinki, Finland.

D. Turcato, O. Laurens, P. McFetridge, and F. Popowich. 1997. Inflectional information in transfer for lexicalist MT. In *Proceedings of the International Conference 'Recent Advances in Natural Language Processing'*, pages 98–103, Tzigov Chark, Bulgaria.

P. Whitelock. 1994. Shake and bake translation. In C.J. Rupp, M.A. Rosner, and R.L. Johnson, editors, *Constraints, Language and Computation*, pages 339–359. Academic Press, London.

D. Wu. 1995. Grammarless extraction of phrasal translation examples from parallel texts. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 354–372, Leuven, Belgium.